Compressão no HTTP

A compressão é uma maneira importante de aumentar o desempenho de um site. Para alguns documentos, a redução de tamanho de até 70% diminui a necessidade de capacidade de largura de banda. Com o passar dos anos, os algoritmos também se tornaram mais eficientes, e novos passaram a ser suportados por clientes e servidores.

Na prática, os desenvolvedores web não precisam implementar mecanismos de compressão, tanto navegadores quanto servidores já possuem isso implementado, mas é necessário garantir que o servidor esteja configurado adequadamente. A compressão ocorre em três níveis diferentes:

- primeiro, alguns formatos de arquivo são comprimidos com métodos otimizados específicos,
- depois, a compressão geral pode ocorrer no nível HTTP (o recurso é transmitido comprimido de ponta a ponta),
- e, por fim, a compressão pode ser definida no nível da conexão, entre dois nós de uma conexão HTTP.

Compressão de formato de arquivo

Cada tipo de dado tem alguma redundância, ou seja, espaço desperdiçado. Se textos podem ter tipicamente até 60% de redundância, essa taxa pode ser muito maior para outras mídias como áudio e vídeo. Ao contrário do texto, esses outros tipos de mídia usam muito espaço para armazenar seus dados e a necessidade de otimizar o armazenamento e recuperar espaço tornou-se aparente muito cedo. Engenheiros projetaram o algoritmo de compressão otimizado usado por formatos de arquivo projetados para esse propósito específico. Algoritmos de compressão usados para arquivos podem ser agrupados em duas categorias amplas:

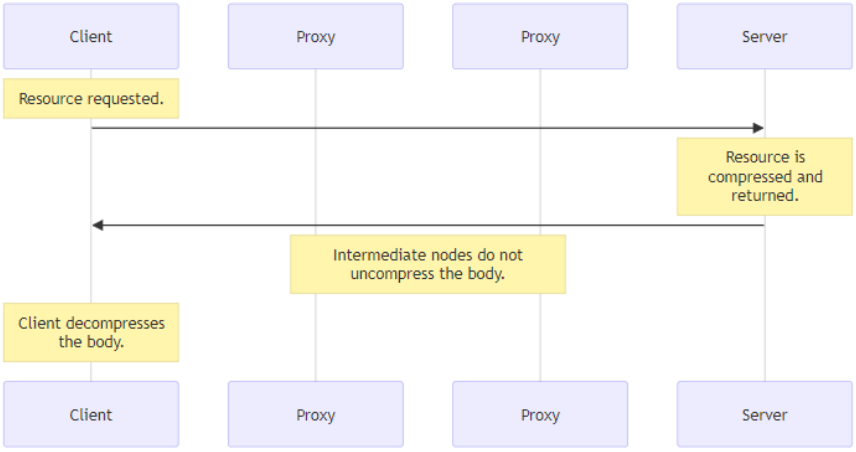
- **Compressão sem perdas**, onde o ciclo de compressão-descompressão não altera os dados recuperados. Ele corresponde (byte a byte) com o original. Para imagens, gif ou png usam compressão sem perdas.
- **Compressão com perdas**, onde o ciclo altera os dados originais de uma forma (esperançosamente) imperceptível para o usuário. Formatos de vídeo na Web são com perdas; o formato de imagem jpeg também é com perdas.

Alguns formatos podem ser usados tanto para compressão com quanto sem perdas, como o webp, e geralmente o algoritmo com perdas pode ser configurado para comprimir mais ou menos, o que, claro, leva a menor ou maior qualidade. Para melhor desempenho de um site, é ideal comprimir o máximo possível, mantendo um nível aceitável de qualidade. Para imagens, uma imagem gerada por uma ferramenta pode não estar otimizada o suficiente para a Web; recomenda-se usar ferramentas que comprimam o máximo possível com a qualidade exigida. Existem inúmeras ferramentas especializadas para isso.

Algoritmos de compressão com perdas geralmente são mais eficientes do que os sem perdas.

Compressão de ponta a ponta

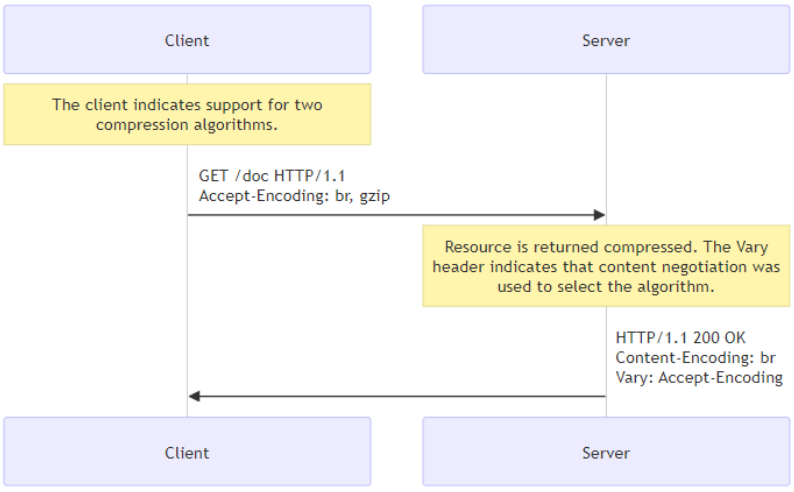
Para compressão, a compressão de ponta a ponta é onde residem as maiores melhorias de desempenho dos sites. Compressão de ponta a ponta refere-se à compressão do corpo de uma mensagem que é feita pelo servidor e permanece inalterada até chegar ao cliente. Quaisquer que sejam os nós intermediários, eles deixam o corpo intocado.



Nota: Como a compressão funciona melhor em um tipo específico de arquivo, geralmente não é útil comprimi-los uma segunda vez. De fato, isso frequentemente é contraproducente, pois o custo do overhead (algoritmos geralmente precisam de um dicionário que aumenta o tamanho inicial) pode ser maior do que o ganho adicional em compressão, resultando em um arquivo maior. Não use as duas técnicas a seguir para arquivos em formato já comprimido.

Todos os navegadores e servidores modernos oferecem suporte a isso, e a única coisa a ser negociada é o algoritmo de compressão a ser usado. Esses algoritmos são otimizados para texto. Na década de 1990, a tecnologia de compressão avançava rapidamente e numerosos algoritmos sucessivos foram adicionados ao conjunto de escolhas possíveis. Hoje em dia, apenas dois são relevantes: gzip, o mais comum, e br, o novo concorrente.

Para selecionar o algoritmo a ser usado, navegadores e servidores utilizam **negociação de conteúdo proativa**. O navegador envia um cabeçalho Accept-Encoding com os algoritmos que suporta e sua ordem de precedência; o servidor escolhe um, usa-o para comprimir o corpo da resposta e usa o cabeçalho Content-Encoding para informar ao navegador o algoritmo escolhido. Como a negociação de conteúdo foi usada para escolher uma representação com base em sua codificação, o servidor deve enviar um cabeçalho Vary contendo ao menos Accept-Encoding junto com este cabeçalho na resposta; dessa forma, caches poderão armazenar as diferentes representações do recurso.



Como a compressão traz melhorias significativas de desempenho, recomenda-se ativá-la para todos os arquivos, exceto os que já estão comprimidos, como imagens, arquivos de áudio e vídeos.

- O Apache oferece suporte à compressão com mod_deflate
- Para o Nginx há o módulo ngx_http_gzip_module
- Para o IIS, o elemento <httpCompression>

Transporte de dicionário de compressão

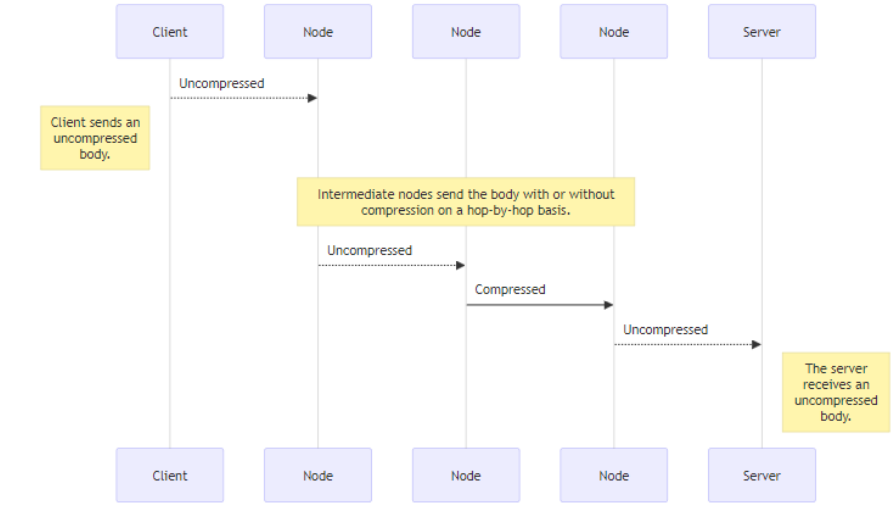
Formatos de compressão modernos como Brotli e Zstandard podem usar dicionários de dados usados com frequência para aumentar ainda mais a compressão, apenas referenciando esses dados a partir do próprio arquivo comprimido.

Tipicamente, para respostas HTTP, isso usa o dicionário estático predefinido incluído naquele formato (por exemplo, o dicionário estático do Brotli está disponível no código-fonte).

Transporte de Dicionário de Compressão permite que um desenvolvedor especifique um recurso que possa ser usado como dicionário para requisições futuras. Isso pode ser um arquivo de dicionário específico, ou um recurso existente (por exemplo, usar app.v1.js como dicionário ao baixar app.v2.js). Isso tipicamente melhora a compressão e, portanto, o tempo de carregamento. No exemplo app.vX.js, a maior parte do download consistiria apenas do delta entre as duas versões, e os bytes em comum poderiam ser referenciados a partir do arquivo app.v1.js que já foi baixado.

Compressão salto a salto

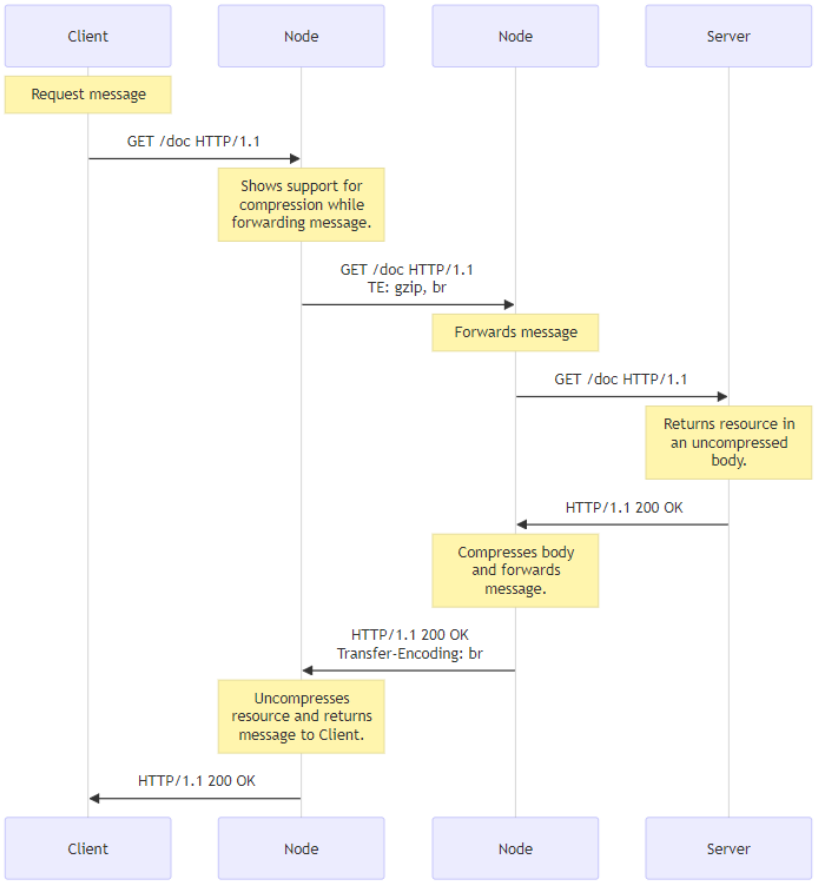
A compressão salto a salto, embora semelhante à compressão de ponta a ponta, difere por um elemento fundamental: a compressão não ocorre sobre o recurso no servidor, criando uma representação específica que é então transmitida, mas sim sobre o **corpo da mensagem** entre quaisquer dois nós no caminho entre o cliente e o servidor. Conexões entre nós intermediários sucessivos podem aplicar compressões diferentes.



- O cliente envia um corpo **não comprimido**.
- Nós intermediários enviam o corpo com ou sem compressão de forma salto a salto.
- O servidor **recebe um corpo não comprimido**.

Representação:

Para fazer isso, o HTTP usa um mecanismo semelhante à negociação de conteúdo para compressão de ponta a ponta: o nó que transmite a requisição **anuncia sua intenção** usando o cabeçalho TE, e o outro nó escolhe o método adequado, aplica-o e indica sua escolha com o cabeçalho Transfer-Encoding.



Na prática, a compressão salto a salto é **transparente para o servidor e o cliente**, e é raramente usada. TE e Transfer-Encoding são usados principalmente para enviar uma resposta em partes (*chunks*), permitindo começar a transmissão de um recurso sem conhecer seu comprimento.

Note que o uso de Transfer-Encoding e compressão no nível de salto é tão raro que a maioria dos servidores, como Apache, Nginx ou IIS, **não possui uma maneira fácil de configurá-lo**. Essa configuração geralmente ocorre no nível de proxy.